THERAC-25

THE INVESTIGATION OF THE THERAC-25 ACCIDENTS.

Vassili Zdanov

Tallinna Tehnikaülikool

Abstract. **This paper is a short analysis of Therac-25 accidents and gives suggestions how to avoid such accidents in the future. Therac-25 is a radiation therapy machine that totally relies on software. In less than 2 years it virulently overdosed six people. Those accidents are known to be the worst in all time medical acceleration history. Many lessons have been learned from the series of accidents and the drawn conclusions are actual even today.**

1. INTRODUCTION

THERAC-25 is a computer-controlled radiation therapy machine capable of producing high-energy beam. It was designed by Canadian AECL and French CGR corporations in 1976. The first time Therac-25 was commercially used was in 1982. The idea was to build a machine capable of producing a high-energy beam that can destroy tumors with minimal impact on the surrounding healthy tissues. To heal a shallow tissue an accelerated electrons are used. In order to reach deeper tissue the X-ray photons are used instead of electrons. This is due to the fact that X-ray has better penetration capabilities. In less than 2 years the Therac-25 massively overdosed six people causing serious injures to patients. The accidents have been described as the worst accidents in the history of medical accelerators. Many information regarding Therac-25 software development, management and quality control are not available and therefore limiting the investigation procedure.

Therac-25, as well as its predecessors Therac-6 and Therac-20, were built by AECL (Atomic Energy of Canada Limited) and French company CGR. The first product was called Therac-6, a 6 million electron volt (MeV) accelerator capable of producing only X-rays. The next machine Therac-20, a 20 MeV, was able to operate in dual-mode producing both X-rays and electrons. The machine was the basis of a newer Therac-25. All Therac machines were computer controlled (DEC PDP-11). The initial software for Therac machines was developed by CGR. In the mid-70s AECL designed Therac-25, dual-mode accelerator delivering photons or electrons at various energy levels. Therac-25 used a new “double-pass” concept that needed much less space to accumulate energy due to the fact that it folds the physical mechanism required to accelerate electrons. Therac-25 was build to be superior to its predecessor, the Therac-20. Therac-25 was more compact and versatile, it was more economical to produce due to the earlier mentioned “double-pass” concept and was easier to use for operators. Besides, Therac-25 was able to deliver various energy levels ranging form 5 to 25MeV.

2. PRINCIPLES OF WORK

The advantage of the higher energy lays on the phenomenon of “depth dose” – the higher the energy, the higher the depth in the body at which maximum dose build-up occurs.

Therac-25 is designed to take advantage of computer control from the outset. The very important issue is that Therac-25 safety relies on software. There were no interlocks or duplication of existing hardware. Therac-20, for instance, had independent protective circuits for monitoring the position of the turntable.

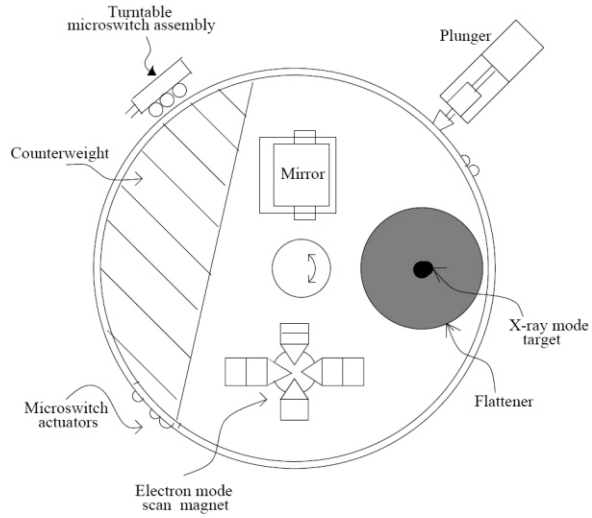

Figure 1: Upper turntable assembly.

Figure 1. Upper turntable assemly

- Turntable. There are 3 main positions of turntable – Electron mode, Photon mode and Field light position to facilitate correct position of the patient. 3 micro switches control the position.
- Scanning magnets are used in electron mode to spread the beam to a safe, therapeutic concentration.
- Computer controls the beam – 5 to 25 MeV for electron mode and 25 MeV for X-ray mode.
- Flattener is used to produce a uniform treatment field which equalizes the x-ray beam intensity.
- Mirror is used for the light simulation of the beam path. It is important for the operator to see precisely where the beam will strike.
- Collimator, a set of movable blocks which shapes the X-ray beam.
- X-ray ion chamber, which measures the strength of the beam.
- Plunger is used to lock the turntable position

If the flattener is not on the appropriate position, an overdose occurs. This is the basic hazard of the dual-mode machines. The computer is responsible for the turntable position so that a target, flattering filter and X-ray ion chamber are on the beam path.

In Therac-25 software checks were substituted for traditional hardware interlocks.

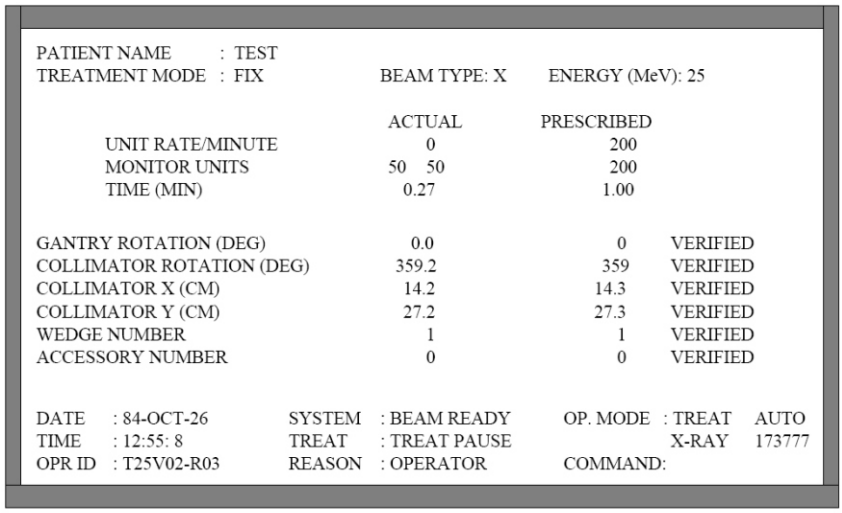

Figure 2: Operator interface screen layout.

Figure 2. Operator interface screen layout

The operator controls the machine through DEC VT100 terminal (Figure 2). An operator enters patient identification, prescription (beam type, energy level, dose, dose rate, and time), field sizing, gantry rotation and accessory data. The system controls the manually set values with the ones entered in the console. If they match, a verified message occurs and treatment procedure can be started.

Operators complained that it took too long to enter the treatment plan. AECL responded by modifying the software so that the operators could use a “Carriage Return” to copy the treatment data. These modifications played an important role in several accidents. The accidents happened due to the fact that during the operation the cursor should be placed over the “Command” line.

There are 2 way for the machine to shutdown:

- Treatment suspend, which required a complete machine reset to restart.
- Treatment pause, which required a single key command to restart. In this case, the operator could press “P” key to proceed and resume the treatment. All parameters left unaffected and no reset required. The feature could be repeated 5 times before the system’s complete reset.

Error messages consisted of the word “MALFUNCTION” followed by a number from 1 to 64. The operator’s manual does not provide any explanations for the malfunction codes. The materials provided give no indication that these malfunctions could place a patient at any kind of risk. Error messages seem to be not abnormal – they appeared on average 40 times a day. Operators were taught that it was impossible to overdose a patient.

3. HAZARD ANALYSES

After series of malfunctions and errors AECL had finally decided to make a hazard analysis. In 1983 they performed a safety analysis in the form of the fault tree and apparently excluded the software.

The report concluded that:

1. Programming errors have been reduced by extensive testing.
2. Does not degrade by time, wear or fatigue.
3. Computer execution errors are caused by faulty hardware components (OR gates, AND gates, etc).

The report didn't contain any detailed information whatsoever. It seemed that AECL performed analyses due to numerous errors that occur daily. The points mentioned above do not seem to be the real cause of the problem. Unfortunately, extensive testing does not reveal software bugs that were the real cause of the malfunctions to occur. Also, it is much less likely that hardware was the reason for those errors. AECL claimed that hardware errors were caused by alpha particles and electromagnetic interference.

AECL has also given probabilities for different errors to occur. For example, for the error in a logical gate (OR, AND, XOR, etc) that will result in selecting a wrong energy was given smaller probability than for the error that will result in selecting a wrong energy mode that could lead to catastrophic results. The real concern was that AECL had never had any justification to these numbers. The probability numbers were so small that it was virtually impossible for the error to occur. It will take years for machine to work to reproduce the fault. As it will be seen later the same error occurred less than in a year, which means that AECL has given random numbers or wrong assumptions.

Unfortunately, AECL has never published their fault-tree analyses.

4. EVENTS

In the following some facts about the accidents:

- A total of 11 Therac-25 machines were installed – 5 in USA and 6 in Canada.
- 6 patients were overdosed between 1985 and 1987.
- 5 patients died because of the consequences of the virulent X-Ray radiation overdose.
- First accidents were not diligently investigated.

The First Accident

The first accident happened in June 1985 at Kennestone Regional Oncology Center. Patient received an estimate of one or two doses of radiation in 15 0000 to 20 000 rad (radiation absorbed dose) range. To understand these numbers consider typical radiation dose of 200 rad to a part of the body. This means that the patient received about 100 times more radiation which led to constant pain. If a dose of 500 rad is delivered to the patients whole body then in 50% of the cases it will cause death. This accident was not investigated. AECL refused to believe that the accident was caused by Therac-25.

The Second Accident

The second accident happened 3 weeks after the first one in July 1985 in Hamilton, Ontario. After the prescription was entered a computer showed "HTILT" message. Operator resumed the treatment by pressing the "P" key. The same message occurred and operator repeated this procedure 5 times before the machine went to the "Treatment Suspend" mode. Patient received an estimate dose of 13 000 to 17 000 rad.

AECL responded that it was a transient failure in the microswitch that determines the turntable position. Manufacturer claimed for design weaknesses in 3 bit signal from microswitches. 1 bit error in the microswitch could produce a wrong positioning of the turntable. AECL installed an additional “In Transit” status of the switches and claimed that this feature improves the system by at least 5 orders of magnitude which equals to 10 000 000 % of improvement over the previous system.

The Third Accident

The third accident happened in December 1985 in Yakima Valley. The reaction was determined not to be abnormal until January 1986 when the patient finished her treatment. The accident was not investigated and AECL reported that the accident could not have been produced by any malfunction of the machine or operator error. There were no further investigations after the accident.

The Forth Accident

The forth accident happened in March 1986 in Tyler, Texas. Patient received an estimate dose of 16 500 to 25 000 rad. The accident had the first real diligent investigation.

Operator typed “x” for X-Ray mode when intended to press “e” for electron mode. Operator used an “UP” arrow key to edit the mode entry; therefore parameter values were left impact by pressing the “Return” key. MALFUNCTION 54 message occurred (Input Dose 2) on the screen displaying “TREATMENT PAUSE”. AECL technician later testified that this error means that the dose was too high or too low. When pressing the “P” command the machine showed the same MALFUNCTION 54 message. By that time the patient was stroked the second time with virulent radiation beam.

AECL started the investigation and suggested the electrical problem in this area.

The Fifth Accident

The fifth accident happened one month later in April 1985 in the same place in Tyler, Texas. The same treatment mode mistake was made as in the previous accident. Operator intended to use electron mode when accidentally typed X-ray mode. Same steps were reproduced as one month earlier. As a result an estimate of 25 000 rad range facial overdose was delivered to patient.

By this time Fritz Hager, Tyler physicist has determined that the data entry speed during the editing was the key factor in producing the error. That led to a dependency of the editing sequence of the Therac-25 which used the same routine as the predecessor Therac-20, but this accident could never take place with Therac-20 because it used a hardware interlocks. After several attempts, AECL could reproduce MALFUNCTION 54 message. It was the first time the manufacturer could reproduce the error and started to investigate the failure in software.

The basic mistakes involved poor software engineering practices and building a machine that relied on software for safe operations. Operator system was a real-time system developed by one programmer in 1970s. Software was written in PDP-11 assembly language that had 4 major components:

1. Stored data.
2. Scheduler.
3. Set of critical and noncritical tasks.
4. Interrupt services.

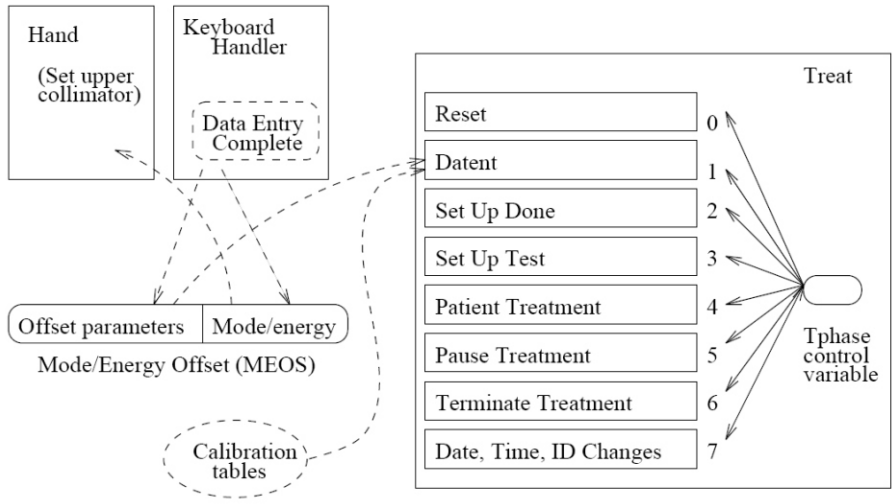

Figure 3. Tasks and subroutines that caused the accident [1]

Treatment Monitor Task (Treat) has 8 main subroutines that control the phase of the treatment. The *Tphase* control variable stores the information to determine which subroutine should be executed. After the execution Treat reschedules itself.

The second subroutine called *Datent* (Data entry) determines if the prescription data is entered. It examines the Data Entry Complete flag. After the variable is set, the *Datent* changes the value in *Tpahse* from 1 to 3 (*Set Up Test*). If the variable is not set, then the *Datent* reschedules itself and does not change the value in *Tphase* and exits back to Treat's mainline.

When operator completes all the necessary steps to the prescription then the cursor is moved to the "Command" line as the normal position (See fig. 2). The *Data Entry Complete* variable does not check that the cursor is located on the "Command" line therefore causing a potential error. In some cases, the data entry phase can exit before all changes are made on the screen. Mode and energy is set through one byte of the *MEOS* variable, which is used to set the proper collimator position for the treatment. If this byte is wrong, then the collimator sets the wrong position.

If the keyboard handler sets the *Data Entry Complete* flag before the operator changes the data in *MEOS* entry, then changes will not affect the *Datent* because it has already exited and will not be reentered. When the parameters are set, the *Datent* calls the subroutine *Magnet* that bends and sets the magnets for their appropriate position. Setting these magnets takes about 8 seconds. The first thing that *Magnet* does is to call a subroutine *Ptime* to make a time delay. Because there are many magnets to be set, the *Ptime* subroutine is entered several times and reschedules itself. When the bending magnets are set *Ptime* is cleared. *Ptime* then checks a shared variable that indicates if any changes have been made. If the variable is set, then edits were made and *Ptime* variable clears the values if the bend magnets and exits to *Magnet* which exits back to *Datent*. But this shared variable is checked only when the bending magnets flag is set. *Ptime* clears the flag during the first execution therefore any changes made during each succeeding pass through *Ptime* will not be detected.

The accident happened in April 1985 when the operator made the entry of energy and mode, moved to the command line then moved the cursor to change the mode and moved back to the command line all in less than 8 seconds that takes the magnets to be set. The editing of the mode has never been detected because *Magnet* does not recognize the edits after the first execution of *Ptime*.

But that is not all. After exiting the *Magnet* subroutine, the *Datent* checks the *Data Entry Complete* flag. If it is set, then *Datent* moves to the next phase *Set Up Test*. If not, *Datent* leaves the *Tphase* without any changes. *Data Entry Complete* flag only checks that the cursor is on the command line, not that it is still there.

AECL fixed those problems. The bending magnets variable is cleared at the end of the *Magnet* subroutine, not *Ptime* subroutine. A new variable was introduced in order to check if the cursor is not positioned on the command line.

Other improvements that have been made to avoid the accidents:

- Food and Drug Administration (FDA) approval.
- "UP" cursor should not be used for editing.
- "R" reset command should be used instead and the whole prescription reentered.
- AECL retained the malfunctioning codes.
- Quality assurance testing (QAS) was done even thou AECL did not intend to make it. FDA insisted on QAS to ensure exact copying of software. FDA has also requested for further rigorous testing for all AECL software modifications. FDA insisted AECL to make an Independent Validation and Verification phase (IV&V), Quality assurance control and

Figure 4. Insisted FDA Quality assurance cycle implemented in Therac-25 software modifications [6]

briefly described safety mechanism for their software modifications (Figure 4).

The Sixth Accident

The sixth accident happened in January 1987 in Yakima Valley. Patient received an estimate dose of 8 000 to 10 000 rad. The accident was diligently investigated and a new software "bug" was detected.

The operator pressed the "B" key for the beam when console displayed "Beam Ready". When the beam came on the display showed no dose or dose rate. After 5 second unit shut itself down with a pause, so that the operator could use "P" command to proceed with treatment. The machine paused again displaying a message "FLATNESS'. By this time patient received a huge radiation dose on the skin.

By this time AECL began to investigate the accident. After careful investigation, AECL engineers found that erroneous machine behavior was not caused by hardware alone. They found a flaw in software that most probably caused the accident and that was totally different from that happened in previous accident. Later, the AECL quality assurance manager said that most probably Hamilton accident can be blamed for the same software error as this one.

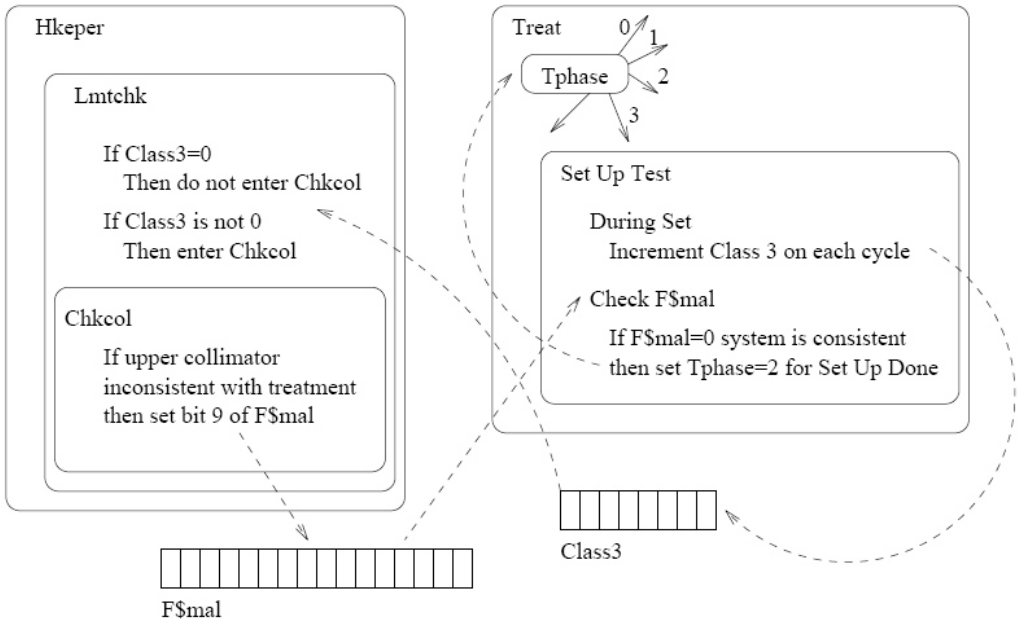

Figure 5. Tasks and subroutines that caused the sixth accident [1]

The prescription is entered at the console after patient has been placed and unverified parameters entered. The same *Datent* routine verifies the data. The next subroutine *Set Up Test* is executed. To check the collimator position, a shared variable *Class3* is set and increments every pass through *Set Up Test* subroutine. *Class3* value should be equal to zero for the treatment to proceed, if not the treatment is not proceeded. If *Class3* variable equals to zero, then *Set Up Test* subroutine calls another shared variable to check if any malfunctions present. The variable is called *F$mal*. In order for the treatment to continue both *Class3* and *F$mal* should be equal to zero. In this case *Tphase* variable is set to 2 and next *Set Up Done* subroutine is executed. If not, *Set Up Test* subroutine is rescheduled.

There are no hardware interlocks in Therac-25. The only software interlock that checks the collimator position is performed by *Lmtchk* subroutine. *Lmtchk* controls both *Class3* and *F$mal* for their value to be equal to zero.

Obviously, *Set Up Test* will be executed many times during the setup because it reschedules itself every time an event occurs. As it was said earlier, the value in *Class3* is incremented every time it passes through *Set Up Test*. This value is 1 byte long which equals to 8 bits and maximum decimal value of 255 (from 0 to 255). That means, that every pass through the value 256 the actual value is set to 0 and that means that at this time the collimator position will not be checked. Since *Set Up Test* is executed hundreds of times during the setup, a couple of checks will not be performed.

The accident happened at the exact moment when the operator pressed “set” command on the 256th check. Consequently, *F$mal* was not set and software turned the X-Ray beam with full energy of 25MeV delivering a massive overexposure to patient.

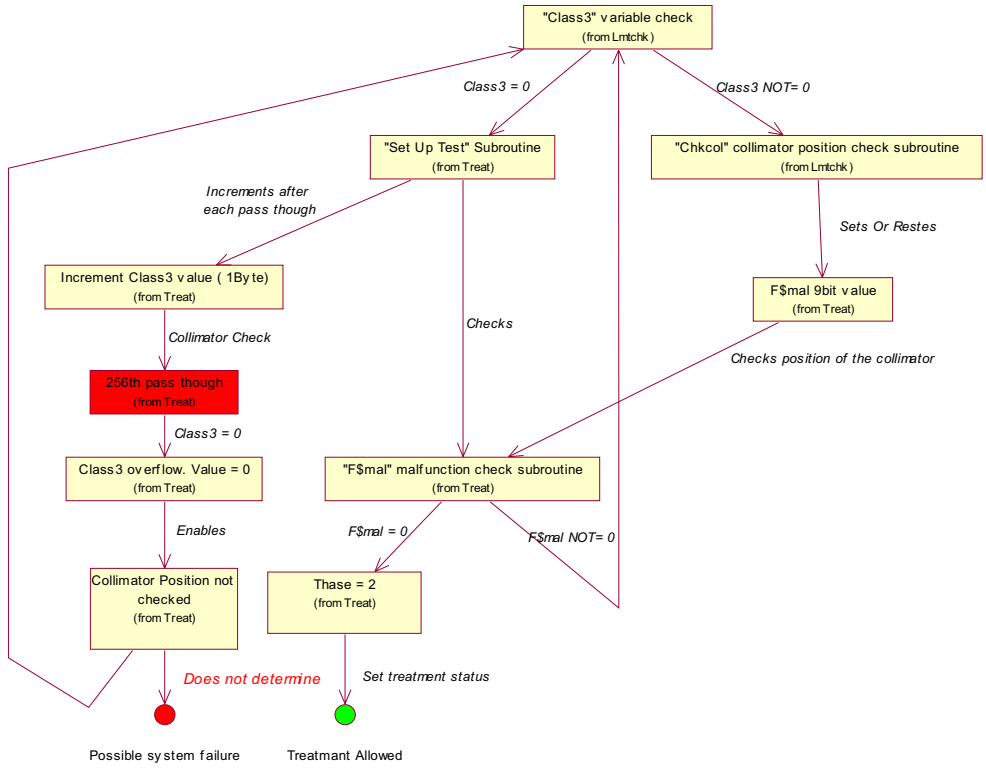

Figure 6. Yakima accident fault tree

AECL fixed the error by changing the *Class3* value to nonzero value. The solution was quite simple to implement.

On July 21, 1987 AECL has announced the final revision that fixed previous accidents and added new featured to improve safety. Most important of them include:

- Interrupt goes to treatment suspend, not treatment pause mode. Re-entering parameters is needed to ensure safety.
- Yakima and Tyler related accidents are resolved.
- Limiting the use of editing keys.
- Malfunction messages will be meaningful.
- A software single-pulse shutdown is added.
- A hardware single-pulse shutdown is added.
- Turntable position monitoring logic to ensure the appropriate turntable position (3 positions).
- Field-light position will not be able to use beam-on.
- A potentiometer added to the turntable.

5. THE ANALYSIS

The fault tree analyses were made before the first accident happened. Unfortunately, AECL apparently excluded the software claiming that it is reliable. This was the main reason for the accidents to happen. The software was reliable enough, but it was not safe. AECL had never followed the rule that safety as a part of the design and development process. No qualitative measures like Safety Integrity Level (SIL) were taken to protect the software against failures. SILs were not defined therefore no safety integrity levels were achieved. It was later obvious that AECL had to make SIL as part of their safety design plan. But, unfortunately, it was ignored.

The FTA (Fault tree analyses) were made on the initial design phase. It identifies a hazard and then tries to show all possible ways to create that hazard. FTA simply answers the question how can an event occur. The first and only AECL FTA was made in 1983, before the accidents.

Formally, AECL FTA could look very similar to example below.

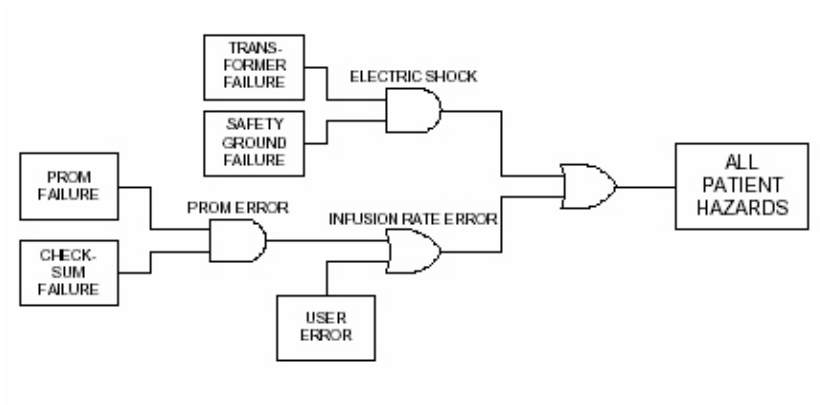

Figure 7. Possible AECL hardware Fault tree after hazard analyses in 1983 [2]

AECL had relied on their software that is why they excluded it from the FTA. AECL engineers had never thought that software could take any place in accidents. This overconfidence showed after the first accidents that were never diligently investigated. After the second accident AECL seem not to be sure about the real reason of the accident, but could only speculate. The hardware microswitch failure seems to be unlikely the reason for the accidents to happen.

After a brief investigation AECL quality assurance manager suspected the same software problem as “Yakima’s bug” to cause the accident in Hamilton. Later, engineers came to a conclusion that all accidents were caused by software flaws. It seemed that AECL had ambiguous opinion about their hardware blaming the faulty hardware components for the first accidents to happen, but on the other hand assuring that hardware to be fine.

A few issues are considered to be the main reason for the accidents to happen. They include:

- Overconfidence in software, which seems to be the main reason for the accidents. The future accidents could have been prevented if AECL had paid more attention to the software.
- Confusing safety with reliability. As it was said before, AECL run thousands of tests assuming their software to be reliable and therefore safe. But designing the software to be testable is absolutely different. This confusion led to software complacency.

- The investigation of root causes. Focusing on particular software design error is not the way to make a system safe. An overall system analysis should have been used. One software “bug” can lead to another one. In this way if hardware is the reason for a catastrophe and should not be relied on, then the same tendency should be kept for the software. Some accidents could never happen with Therac-20 because it used hardware interlocks. A right design uses both – software and hardware safety mechanisms and interlocks.
  One of the biggest mistakes made is the confidence that the cause of the accident had been determined. It was a wrong tendency because the first accidents could never be reproduced (even though AECL believed it was a hardware problem). Fixing each individual software error does not make the system safe and solve all the safety problems of Therac-25.
- Weak defensive design. Not even Therac-25 did not have any hardware interlocks; the software design was poor for error-handling and self-checks. There was no detection for the overdose to occur. Patient’s reactions were the only indication of the problem. The possible solution was to add a detection interlock or routine when the overdose occurs. This could have been done with independent subsystems.
- Software reuse. It is a wrong assumption that software is safe and reliable because it was used extensively before. Therac-25 used a Therac-20 software design. AECL designers added more complexity that led to a difficulty of the integration. Therac-25 was more complex and totally relied on software. In many cases it is far safer to write a clean and simple design from the list.
- Poor software engineering practices. Some mistakes were made during the software design. This includes the lack of documentation and explanation to malfunctions. Operators were only told that the machine was highly reliable. The malfunction numbers could have been explained in users manual as well as the risk factor that this malfunction could bring to patient. Lack of quality assurance practice has prayed its role. Also, regression and extensive testing should have been run. Most of the time Therac-25 run integrated system test almost bypassing unit and software testing.
- Response to accidents. As it was mentioned before, the first accidents were never investigated. It is wrong position of AECL. Diligent investigation should have started after the first accident in Kennestone.
- Complacency. It is bad, but sometimes an accident is required to warn about the danger involved in technology. Before Therac-25 there were almost no accidents at all. But after the accidents the reputation of software being fault-free seems to be spoiled.
- Untrue risk judgment. As it was mentioned before, the hazard analyses performed by AECL excluded the software. There was a wrong probabilistic risk assessment generated. The third accident was never investigated because AECL assured users that they improved the machine by 5 orders of magnitude meaning by 10 000 000 % over the previous version. AECL assumed that any errors could simply not happen, especially with software. The typical problem is that these kinds of analyses exclude aspects of problem that have much deeper impact on the problem.
- Friendly user interface. Making user interface simpler is better. It is easier to use for operators and eliminates multiple data entry and careful overview of the values entered.
- Government standards. It is very important for government and federal agencies to be involved. FDA made a big pressure on AECL making the investigation procedure much faster and productive. Government should always support social projects.

6. CONCLUSION

Many lessons can be learned from the accidents happened with Tharac-25. It seems that Therac-25 is a classic example on how the system was not build appropriately and testable.

Issues that describe the reasons for the accidents to happen were already mentioned. Unfortunately, design mistakes led to system failures which considered being the worst accidents happened with medical accelerators in all time.

The most important fact to be learned from the series of accidents is to build the system appropriately and testable in all fields. It is not acceptable for the potential health hazard systems to ignore testability.

REFERENCES

[1] N. Leveson, “Medical Devices: The Therac-25”, University of Washington

[2] E. Jaffe, “Developing Medical Software: Pitfalls and Prophylactics”, Seminar in computer assisted-surgery, medical robots and medical imaging, 2002

[3] N. Leveson, C. Turner, “An investigation of the Therac-25 accidents, IEEE Computer, Vol. 26, no.7, 1993

[4] “Therac-25 case materials”,
http://computingcases.org/case_materials/therac/therac_case_intro.html

[5] J. Lim, “An engineering disaster: Therac-25”,
http://www.bowdoin.edu/~allen/courses/cs260/readings/therac.pdf, 1998

[6] Jason G. Cooper, Keith A. Pauley, ”Healthcare Software Assurance”,
http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1839424, Mid-Atlantic Technology, Research, and Innovation Center, South Charleston, WV, USA

KOKKUVÕTE

Käesolev artikkel on lühike analüüs meditsiinilise radiatsiooniseadmega Therac-25 toimunud õnnetustest ning annab soovitusi, kuidas taolisi õnnetusi saaks tulevikus vältida. Therac-25 on esimene radiatsiooniteraapia masin, mis täielikult põhineb tarkvaral. Vähem kui kahe aasta jooksul andis masin kuuele patsiendile väga suure kiirituse üledoosi, mille tagajärjel patsiendid kas surid või said tõsiseid vigastusi. Need õnnetused on jäänud seniajani kõige tõsisemateks meditsiinilise radiatsiooni õnnetusteks ning kirjeldatud juhtumitest on palju õpitud, kuid õnnetuste analüüsist tulenevad järeldused on kindlasti aktuaalsed ka tänapäeval.

TERMINID

Mõiste inglise keeles	**Mõiste eesti keeles**
Radiation therapy	Radiatsiooniteraapia
Accelerator	Kiirendi
1 eV = 1.602 176 53(14)×$10^{?19}$ J	eV – elektronvolt
Interlock	Blokeering
Turntable	Pöördlaud
Collimator	Kollimaator
Plunger	Varbkolb
Gantry	Estakaad
Malfunction	Häire
Fault tree	Rikkepuu
Alpha particles	Alfaosakesed
Electromagnetic interference (EMI)	Elektromagnetiline häire
Radiation absorbed dose (rad). 1 rad = 0,01 Gy (Gray).	Neeldunud doos
Transient failure	Korrapäratu rike
Regression testing	Regressioonitestimine
Unit testing	Komponenditestimine

Vassili Zdanov
Tallinna Tehnikaülikool
Infotehnoloogia teaduskond